

## **PROYECTO: CONSTRUCCIÓN DE SOFTWARE**

### **1. Objetivos**

Esta práctica tiene la finalidad de:

- Aplicar una metodología de construcción de sistemas orientados a objetos.
- Analizar diferentes alternativas de modelado.
- Utilizar diferentes herramientas de diseño.
- Trabajar a partir de los requerimientos de una aplicación real simplificada.

### **2. Metodología y evaluación**

- Desarrollo de un proyecto dividido en 2 partes obligatorias de presentación e informe.
- Trabajo en equipos de número reducido de alumnos.
- Cada equipo de desarrollo estará encargado de la especificación, diseño, implementación y documentación, de manera independiente en principio, de los sistemas desarrollados por los otros equipos, por lo que deberá distribuir y administrar adecuadamente los tiempos y habilidades de sus integrantes.
- Cada equipo deberá extender las especificaciones del sistema de forma que su proyecto sea distinto al de los demás equipos.
- Cada equipo elegirá (o le será asignada) un área de interés diferente al de los demás equipos.
- Cada equipo especificará y entregará una versión propia completa del Sistema, por lo que podrá extender, en acuerdo con los docentes a cargo, la definición del sistema descrita en el ítem 6.
- El desarrollo del proyecto sigue una metodología OO, adaptada al proceso de aprendizaje (para obtener una idea más detallada ver ítem 7: Anexo TP).
- Confección de una Carpeta por equipo, con la documentación del sistema conforme a las directivas del presente (ítemes 1 a 5) y de las dadas por la cátedra de manera complementaria.
- La documentación a entregar debe estar íntegramente desarrollada mediante las aplicaciones y herramientas que sean convenientes.
- La arquitectura del modelo debe proponerse sobre la base del principio de N-Capas, en especial separando la capa de interfaz de la capa de dominio.
- Construcción de un Prototipo No funcional (aunque navegable y con datos ficticios, orientado al usuario) con interfaces (pantallas, reportes, correo, etc.) de diseño profesional.
- Utilización del laboratorio informático.
- Presentación, discusión y síntesis grupal, en diferentes momentos de la construcción.
- Desarrollo y ejemplificación en clase con aporte de material bibliográfico, referencias y apuntes de todos los temas requeridos.
- Los docentes a cargo del curso definirán el nivel mínimo a alcanzar en cada presentación por el equipo de trabajo en cuanto a Documentación, Participación individual y/o grupal, Plazos, etc.
- Es conveniente que la Carpeta Grupal esté disponible para trabajo y revisión en todos los momentos del ciclo lectivo en que sea requerida.
- Los plazos de entrega son fijados durante el ciclo lectivo. Una vez indicados, salvo casos de fuerza mayor y grupal, no son prorrogados.

- Una vez finalizado y aprobado el trabajo, cada integrante del grupo deberá asegurarse de mantener una copia de la carpeta grupal completa.
- La calificación del primer parcial se obtiene a partir de la evaluación del informe con el Modelo de Requisitos y la participación observada.
- La calificación del segundo parcial se obtiene a partir de la evaluación de la aplicación resultante y la documentación asociada presentada.
- Es exigencia especial de la cátedra la aprobación (en completitud, correctitud y consistencia) del segundo parcial o su instancia recuperadora a los efectos de obtener la Regularidad en la asignatura. En caso contrario se deberá Recursar la misma.
- La situación particular de No regular se dará cuando el nivel alcanzado en la aplicación y su documentación sean correctos y consistentes, pero la completitud no haya alcanzado lo indicado. Algunas particularidades de los requerimientos que “pueden” permanecer incompletos parcialmente se indican oportunamente durante el curso lectivo.
- Para rendir el examen final es condición que la carpeta del proyecto se encuentre aprobada y firmada por uno de los docentes a cargo, o se haya otorgado la autorización correspondiente por medio indicado durante el curso lectivo.
- La plataforma de desarrollo y explotación elegida responde a:
  - Servidor Web: Apache, IIS u otro
  - Lenguajes fundamentales: HTML, PHP y Javascript
  - Lenguaje y componentes opcionales: Applets Java, Flash Presentations.
  - Motor de Base de Datos: MySQL inicialmente. En un futuro, se piensa adquirir otro motor, por lo que se solicita que el sistema actual facilite la migración.
  - Servidor de Correo: si bien la cátedra propone algunos durante el cursado queda a elección de cada equipo.
  - IDEs y Herramientas de Diseño: si bien la cátedra propone algunos durante el cursado queda a elección de cada equipo.

### 3. Características Generales del Informe Final

- Una carpeta por grupo conteniendo la documentación impresa completa.
- Debe numerarse cada folio del informe con un número consecutivo (aunque no es adecuado puede ser colocado manualmente)
- A partir de la segunda hoja, cada una debe presentar :
  - ✓ Encabezado: Programación Avanzada, TP1, Número de Hoja
  - ✓ Pie: Fecha, Apellidos y Nombres de Integrantes del grupo.
- Se debe identificar debidamente cada anexo y deben estar claramente indicadas las referencias a los mismos.
- Se debe mantener la estructura y el estilo del informe, en particular la numeración y el título de cada ítem.

### 4. Estructura del Informe Final

- Primera Hoja o Portada de la carpeta donde figuren :
  - ✓ Integrantes del Grupo.

- ✓ Identificación de la Cátedra, Profesores a cargo, Institución, Año Lectivo.
- Segunda Hoja: En la parte superior deben figurar el nombre, apellido y firma de cada integrante del grupo así como un título que diga **Observaciones**, dejando el resto de la hoja en blanco
- Tercera Hoja: Índice de la Carpeta (actualizado)
- Hoja siguientes: Modelo de Requisitos:
  - ✓ Objetivo general y específicos
  - ✓ Alcances del Sistema explicitado a través de Tablas de Referencias de Funciones y Atributos del Sistema.
  - ✓ Listado de los supuestos, simplificaciones y restricciones del sistema.
  - ✓ Glosario
  - ✓ Tabla de Actores
  - ✓ Diagrama Global de Casos de Uso
  - ✓ Listado de Casos de Uso
  - ✓ Tabla de Clases Potenciales con Justificación de Clases Candidatas
  - ✓ Modelo Conceptual del Dominio
- Hoja siguientes: Modelo de Análisis + Diseño:
  - ✓ Por cada caso de uso resultante (en los ítemes con \* ordenar por subflujo o subcaso de uso si corresponde):
    - Caso de Uso Detallado (considerando una descripción de caso de uso real desde una perspectiva externa)
    - Interfaces Usuarías asociadas \*
    - Diagramas de Secuencia (considerando una representación del caso de uso real desde una perspectiva interna) \*
  - ✓ Diagramas de Transiciones y Estados para los objetos fundamentales.
  - ✓ Diagrama de Clases de Diseño (único e integrado)
  - ✓ Diccionario de Clases Simplificado (Listado completo de clases discriminadas por capa y Detalle de cada clase con Nombre-Atributos-Métodos-Relaciones de cada clase y para los casos no evidentes su Descripción)
  - ✓ Diagrama de Actividad del workflow esencial del sistema mostrado en Ncapas.
  - ✓ Diagrama de Componentes
  - ✓ Diagrama de Clases de Diseño Parcial (al aplicar patrones en acuerdo con prof.)
- Anexos al informe: Estos lineamientos + Documentación complementaria + Registro de citas bibliográficas y/o material de investigación utilizado.
- Última hoja: Un sobre adherido, conteniendo el resultado del desarrollo en soporte magnético (CD o disquetes). Este contenido responde a:
  - ✓ Archivo único con un contenido idéntico al informe presentado de manera impresa, en formato MSWord.
  - ✓ Archivos con el código-fuente resultante de la implementación.
  - ✓ Archivos de copia directa de la base de datos

- ✓ Archivo dump de la base de datos en formato de intercambio
- ✓ Directorio Backup conteniendo una copia de todo lo anterior.
- ✓ Eliminar todos los archivos innecesarios (de prueba, accesorios, duplicados, etc.) al entregar la información en el soporte correspondiente

## 5. Caso de Estudio: “Help Desk”

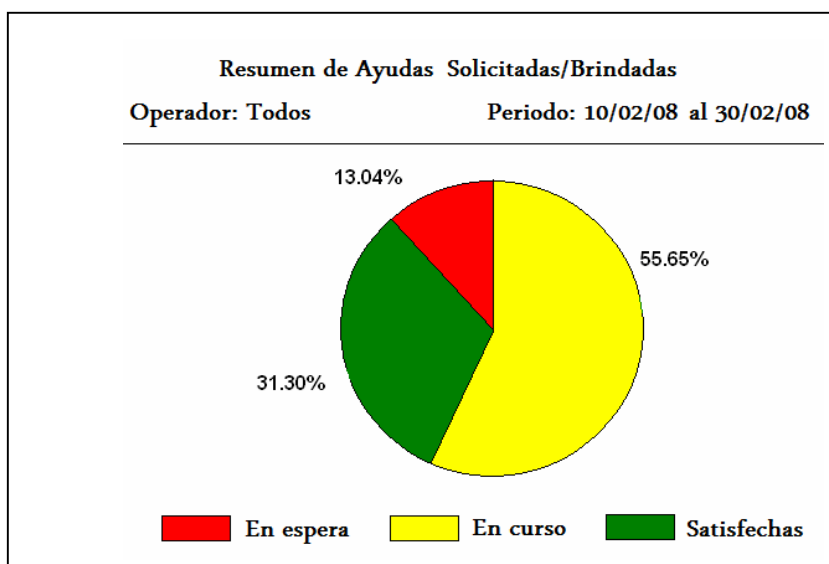
### a. Generalidades

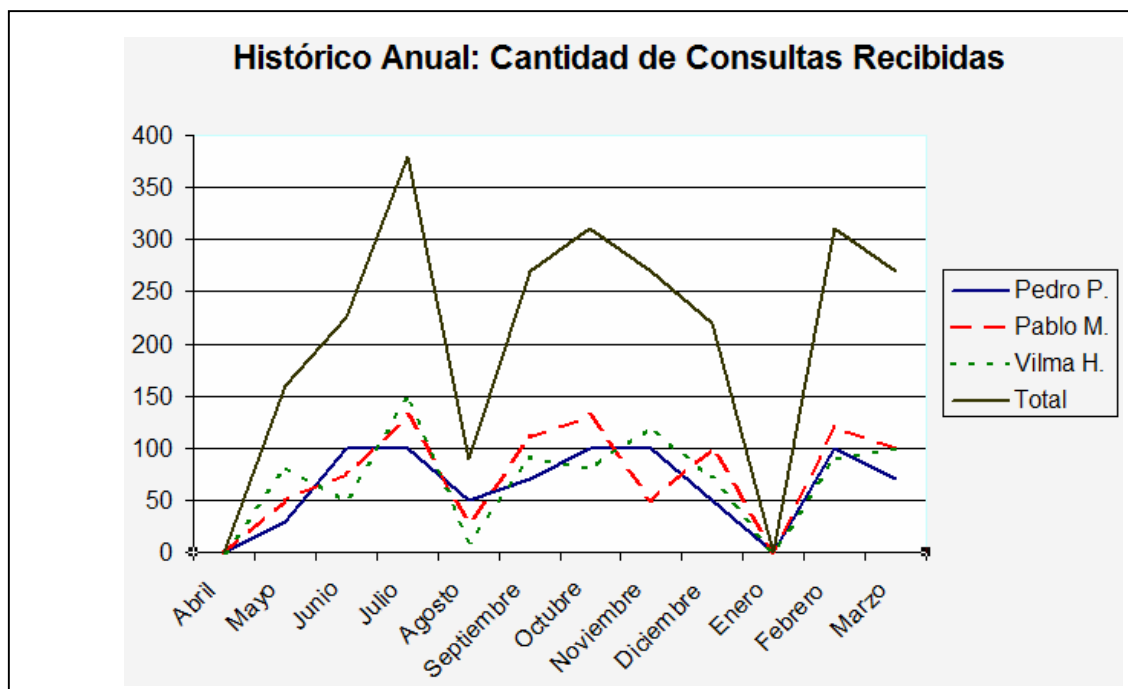
El Sistema de Mesa de Ayuda (SMA) debe cumplir las siguientes características mínimas:

- Debe contar con un sistema de registración de usuarios.
- Cada usuario debe estar definido de manera unívoca en el sistema.
- Se permite a los usuarios darse de alta en el sistema de manera automática, a partir de un formulario de registro.
- La autorización para acceder al sistema se produce a partir de la corroboración de los datos del cliente por parte del operador. En este momento se envía a la dirección de correo válida provista por cada usuario la clave de acceso. Cada usuario debe registrar de manera obligatoria al menos 1(una) dirección de correo.
- Debe permitir trabajar a diferentes usuarios, validando y manteniendo el perfil de cada uno.
- Se debe proponer un portal general donde se ubique el punto de acceso al módulo de ayuda.
- Cada usuario puede en cualquier momento modificar sus datos personales.
- Las consultas pueden ser hechas en tiempo real (interactiva) o diferido. En el primer caso se opera mediante un sistema de mensajería/Chat, mientras que el segundo lo hace mediante mensajería/Correo. Ambos mecanismos son internos al sistema en discusión.
- En el caso que las consultas no puedan ser satisfechas en el momento pueden dejarse señaladas como “En espera”.
- Las consultas pueden ser clasificadas como de “Servicio Técnico”, “Informe Administrativo”, “Error”, “Otros”.
- En las consultas el cliente puede enviar al operador fotos o imágenes descriptivas de su problema.
- Cada usuario cliente debe tener un historial y cierta categorización.
- El operador debe poder realizar consultas al sistema bajo diferentes criterios de búsqueda y filtrado
- Los reportes deben presentarse bajo diferentes criterios de orden (por ejemplo: por fecha, alfabéticamente por descripción, por tipo de consulta, por área del problema, etc.)
- Los reportes resultantes de una consulta deben presentarse de manera paginada, en bloques de información aceptables, sin embargo cada usuario puede decidir modificar la cantidad de ítemes desplegados por pantalla, para una consulta particular o mantener dicha parametrización en su perfil particular de usuario.
- En el esquema de trabajo de la Organización puede haber varios operadores trabajando de manera simultánea.
- Se debe mantener un log del trabajo de cada operador.

- Existe un auditor capaz de consultar sobre las actividades de uno o varios operadores, así como gráficas estadísticas (mostradas en el punto b de este ítem).
- Se deben permitir sólo las acciones que correspondan al perfil del usuario logueado.
- Si un usuario se encuentra realizando altas, bajas o modificaciones debe cuidarse (dentro de los límites admitidos) la presentación de un mensaje de alerta, advirtiéndole que no ha salvado y permitiéndole que salve o descarte los cambios hechos, cuando sea conveniente.
- Se debe contar con ayuda sensible al contexto de operación, aunque no es necesario presentar en la primera versión del sistema. Se deberán presentar al usuario todos los mensajes de error, advertencia o ayuda que sean necesarias, atendiendo a que los usuarios potenciales desconocen el sistema.
- El mecanismo solicitado para proveer seguridad básica es mediante gestión de sesiones, claves encriptadas en la base de datos y de ser viable encriptación adicional en las interfaces clientes.
- No se exige en este trabajo el uso de protocolos seguros, mecanismos de cifrado o firma digital. Su aplicación es opcional.
- El sistema está destinado a ser usado sobre una intranet en un ambiente Web multiusuario y con almacenamiento centralizado.
- Las Bases de Datos podrán ser creadas y/o modificadas mediante un manejador de bases de datos convencional. No se solicita para este sistema la gestión de estructura de la misma.
- Se desea particularmente que se construya una interfaz gráfica amigable y eficiente, de estilo similar a herramientas ofimáticas estándares.
- El diseño de las páginas y el despliegue de imágenes en las páginas deben hacerse de modo tal de optimizar la velocidad de respuesta del sistema.
- El diseño de las interfaces debe proveer una impresión profesional y de ser posible mediante estilos que lo hagan fácilmente modificable.
- El rubro al cual está dedicado este sistema es acordado entre el grupo y el docente.

**b. Anexo de Gráficas mínimas esperadas**





**c. Extensiones**

- Sería conveniente buscar, recopilar y adjuntar a la documentación del sistema todo material y referencia bibliográfica o enlace de Internet donde sea posible encontrar información que justifique o aclare las consideraciones de cada grupo en particular, referido a cualquiera de los aspectos considerados en este Trabajo.
- Cada grupo podrá proponer mejoras adicionales al producto final (sujeto a autorización).

## 6. ANEXO TP : Metodología de Desarrollo OO

Este anexo se incluye a modo descriptivo, no normativo.

Durante el desarrollo de este práctico pueden seguirse diversas metodologías, y cada grupo puede asumir una en particular, previo acuerdo con los docentes a cargo.

Como se aclaró, se trata de una metodología orientada a objetos iterativa (en sus fases y actividades), incremental (en los resultados obtenidos), centrada en la arquitectura (posible MVC) y conducida por casos de uso.

Se muestra a continuación un listado secuencial aproximado con las actividades seguidas durante el ciclo lectivo:

### ★ Primera Iteración y Modelo de Requisitos

1. Definir el objetivo general y al menos 3 objetivos específicos.
2. Analizar, ajustar y determinar las Funciones y Atributos del sistema.
3. Analizar, ajustar y determinar los riesgos y las limitaciones.
4. Identificar y describir los Actores del Sistema
5. Identificar y describir el comportamiento esencial en un Diagrama de Casos de Uso mediante UML.
6. Realizar las Descripciones de Casos de Uso de Alto Nivel.
7. Iniciar el documento de la Especificación de Requisitos del Software (SRS) en base a lo anterior, incluyendo además un glosario base.
8. Tomar nota de los escenarios de cada Caso de Uso, así como también de las pre y post condiciones de cada caso de uso y del ejecutor.
9. Documentar los casos y sus escenarios en formularios de caso de uso detallado.
10. Identificación de las Clases candidatas del Modelo de Dominio y llenado de la Tabla de Clases.
11. Depurar la identificación y representar el Modelo Conceptual del Dominio utilizando UML.
12. Iniciar el Diccionario de Clases (simplificado a : Nombre-Descripción de cada clase)
13. Diseñar las interfaces de usuario asociadas a cada caso de uso (si corresponde)
14. Ajuste del SRS.

### ★ Segunda Iteración y Modelo de Análisis

15. Para cada caso de uso, identificar las abstracciones clave mediante el análisis de estereotipos borde, entidad y control. NOTA: Si bien el modelo de arquitectura básico es MVC, se aclara que puede ser cambiado por cada grupo en acuerdo con los docentes.
16. Elaborar una Tabla multicapa de Identificación de Clases, según estereotipos borde, entidad y control, por caso de uso.
17. Analizar el workflow esencial del sistema y representarlo en un Diagrama de Actividad del sistema combinado con un modelo de capas (que permita identificar la vinculación entre las capas de Presentación y Negocio básicas así como la secuencia de trabajo) [unir y plegar las hojas con el diagrama resultante]
18. Identificar y representar los elementos fundamentales de interacción entre objetos en Diagramas de Secuencia UML para cada caso de uso.
19. Actualizar el Diagrama de Clases del Dominio.

20. Actualizar el Diccionario de clases de ser necesario.
21. Actualizar los casos de uso de ser necesario.
22. Actualizar el Glosario, así como los supuestos y restricciones.

★ Tercera Iteración, Modelo de Diseño

23. Definir Subsistemas y representar mediante Diagrama de Componentes
24. Determinar prioridades de implementación, definir la gestión de almacenamiento.
25. Ampliar el modelo de Dominio por cada caso de uso. [opcional]
26. Incorporar tipos de los Atributos
27. Traducir las Asociaciones, las Agregaciones y las Clases de Asociación
28. Analizar los estados de los objetos fundamentales mediante diagramas de Transiciones y Estados.
29. Perfeccionar los métodos del modelo de Dominio. [opcional]
30. Generar las vistas de los diagramas de secuencia del modelo de Diseño.
31. Actualizar el Diagrama de Clases del Dominio [en caso de ser necesario, unir y plegar las hojas con el diagrama resultante]
32. Actualizar Diccionario de clases, incorporando atributos, métodos y relaciones.
33. Sincronizar todos los artefactos y elaborar un documento único de Diseño del Sistema.

★ Cuarta Iteración, Modelo de Diseño Final e Implementación

- ★ Aplicación de patrones de diseño al Modelo de Diseño Solución [opcional]
- ★ Diseño e implementación del modelo de Almacenamiento específico (con eventual uso de patrones de almacenamiento).
- ★ Ajuste de las interfaces de usuario.
- ★ Actualizar el Diagrama de Clases del Dominio [en caso de ser necesario, unir y plegar las hojas con el diagrama resultante]
- ★ Implementación final y prueba del modelo propuesto
- ★ Re-sincronizar todos los artefactos y ajustar el documento único de Diseño del Sistema